Vol. 14 No. 1 天 文 研 究 与 技 术 第 14 卷 第 1 期
Jan., 2017 ASTRONOMICAL RESEARCH AND TECHNOLOGY 2017 年 1 月

CN 53-1189/**P** **ISSN** 1672-7673

基于 ROACH2-GPU 集群相关器的研究*
——F-engine 模块的设计与实现

牛晨辉[1,2]，汪群雄[1,3]，郑小平[2]，田海俊[1,3]，吴锋泉[1]，李吉夏[1]，陈学雷[1]，蒿 杰[4]

(1. 中国科学院国家天文台，北京 100012；2. 华中师范大学，湖北 武汉 430079；
3. 三峡大学，湖北 宜昌 443002；4. 中国科学院自动化研究所，北京 100190)

摘要：相关器在射电天文中具有重要作用。以往的相关器多采用现场可编程门阵列(Field-Programmable Gate Array，FPGA)或者特定用途集成电路(Application Specific Integrated Circuit，ASIC)技术，开发周期长，不便扩展和改进升级。近年来许多新研制的射电干涉阵相关器采用具有通用架构的现场可编程门阵列和图形处理器的相关器。针对暗能量射电探测实验(天籁计划)的需求，开发了一套基于可重构开放架构计算硬件(Reconfigurable Open Architecture Computing Hardware，ROACH2)和图形处理器的异构相关器，将相关器的数据采集、快速傅里叶变换(Fast Fourier Transform，FFT)等功能与复数乘累加运算功能分开，充分利用了现场可编程门阵列的硬件资源和图形处理器的运算速度。该相关器易于扩展，且运算负载可根据实际运算能力进行不同节点的分配，非常灵活。目前已经应用到天籁计划项目中。

关键词：相关器；F-engine；ROACH2；天籁计划

中图分类号：P111.47 **文献标识码**：A **文章编号**：1672-7673(2017)01-0060-10

射电天文学中，相关器主要用于将不同天线信号做相关运算得到最终的可见度(Visibility)，可见度中包含了相关所得到的射电干涉信息，可由其重构出天图[1]。相关运算是信号处理中常用的一种方式，可描述两个信号在不同时刻的相关程度。处理过程可分为两类，一类是先计算相关函数再做复数傅里叶变换到频域空间，即 XF 型相关。另一类相反，先将信号转换到频域空间，然后在频域空间做复数相乘，即 FX 型相关。由于频域相乘运算与时域卷积等价，所以得到的结果一样。但是随着相关路数 N 的增加，FX 算法要比 XF 算法操作容易很多，这是因为 XF 相关要求 N^2 步快速傅里叶变换算法，而 FX 只需要 N 步[2]。当天线阵列较大，对计算能力的要求很高时，利用 FX 相关器更合适。

随着射电技术的快速发展，干涉阵列数目越来越多，干涉阵列也越来越大。例如天籁计划[3]中，一期已经拥有 96 个双极化阵元(192 路信号输入)，而二期阵元数将增加到近千个；再如国际合作的平方千米阵列(Square Kilometer Array，SKA)项目，一期中频阵具有约 200 个碟形天线，未来可能扩展至几千个阵元。这对相关器的扩展性、可移植性和运算能力的要求越来越高。目前国际上大部分干涉阵中相关器普遍采用特定用途集成电路和现场可编程门阵列硬件，该类相关器一般针对特定的干涉阵设计，不便实现在其他射电阵列的移植。而且如果未来望远镜进行升级，对于传统的现场可编程门阵列和特定用途集成电路也相当困难，往往需要完全重新开发。

根据以上需求开发了一套基于 ROACH2-GPU 架构的相关器。该套相关器采用 FX 型相关，从功能上分为 F-engine 和 X-engine 两部分。F-engine 主要将模拟信号通过模拟数字转换器(Analog to Digital Converter，ADC)转化为数字信号，并进行快速傅里叶变换。X-engine 的主要功能是做同一频点不同通道的复数相加相乘运算得到能见度，最终将数据存盘。该套相关器有较好的扩展性，随着天线阵扩

* 基金项目：国家自然科学基金 (U1231123，11503012，U1331202，U1431108，11473044)；国家高技术研究发展计划 (863 计划) (2012AA121701) 资助.

收稿日期：2016-05-10；修订日期：2016-05-28

作者简介：牛晨辉，男，博士. 研究方向：射电天文信号处理. Email：peterniu@nao.cas.cn

通讯作者：田海俊，男，副教授. 研究方向：天文信息学. Email：hjtian@lamost.org

建，相关路数增加，只需要增加相应的硬件设备即可。相关器中用于实现快速傅里叶变换的部分(F-engine)基于由 U. C. Berkerley 的天文信号处理和电子学研究合作组(Collaboration for Astronomy Signal Processing and Electronics Research，CASPER)研发的 ROACH2 平台开发，这种平台在国际射电天文界应用广泛，用户较多，有较丰富的开发库。目前国际上已经有很多干涉阵采用这个平台开发相关器 F-engine，如探测黑暗时期的大孔径实验(Large-Aperture Experiment to Detect the Dark Ages，LEDA)，探测再电离时期的精确阵列(Precision Array for Probing the Epoch of Re-ionization，PAPER)[4]。相关器 F-engine 模型有较好的移植性，对于不同的干涉阵列，所需要采样频率或快速傅里叶变换长度不一样，采用 ROACH2 开发平台可以较方便地将本模型移植到其他射电干涉阵列中。

考虑到信号采集时所用的现场可编程门阵列本身已有相当的计算资源，为了充分利用这些资源，同时又能解决大规模阵列的高强度计算问题，设计中采用 ROACH2-GPU 异构模式，将采集、快速傅里叶变换运算任务分配给前端 ROACH2，而将算法简单但在大规模阵列中计算强度大的乘累加运算分配给高性能图形处理器，从而提高了相关器的整体性能。这种异构的相关器可以充分发挥现场可编程门阵列和图形处理器各自的优势。本文主要对 F-engine 设计的整体架构和测试结果做进一步的介绍。

1　F-engine 模型架构设计

该相关器的 F-engine 基于可重构开放架构计算硬件开发。天文信号处理和电子学研究合作组提供了基于 ROACH 的图形可视化编程设计[5]。开发库较丰富，提供了如模拟数字转换器、10 Gbe 等硬件端口映射模块，还有如逻辑判断、快速傅里叶变换等常用现场可编程门阵列算法模块，底层库模型配有详细说明，比较便于搭建射电天文后端如相关器和频谱仪等[6]。ROACH2 是 ROACH 的改进版本，它继承了 ROACH[5] 的功能并在硬件上做了提升。ROACH2 平台满足大部分射电天文硬件开发的需求，在射电领域开发更具有通用性。

F-engine 在相关器中主要将模拟信号通过模拟数字转换器转化为数字信号，并进行快速离散傅里叶变换，再将数据从通道-频点转换成频点-通道的形式，最终将数据传输到 X-engine。传统的 F-engine 设计集成程度很高，多路信号通道的 F-engine 功能可能在同一块或几块现场可编程门阵列中实现。这样的 F-engine 虽可满足当前设计的需求，但不够灵活。为了有更好的扩展性，开发了更为通用的模型。在设计的相关器 F-engine 中，每块 ROACH2 相当于一个 32 通道的子 F-engine，通过网络传输将 6 块 ROACH2 数据拼接起来，最终完成 192 通道的 F-engine，且每块 ROACH2 运行相同的模型，在硬件上并无差别，仅通过软件控制，在发送数据包中加入相应参数区别不同 ROACH2 发来的数据。这样可以根据需求拼接出任一 32 倍通道数的相关器 F-engine，易于扩展。

相关器 F-engine 继承了探测再电离时期的精确阵列[4]相关器模型的整体架构，由模拟数字转换器、多相滤波器(Polyphase Filter Bank，PFB)、通道均衡器(Equalizer)、数据转置(Transpose)、以太网(Ethernet)等主要模块构成。整体架构功能如图 1。

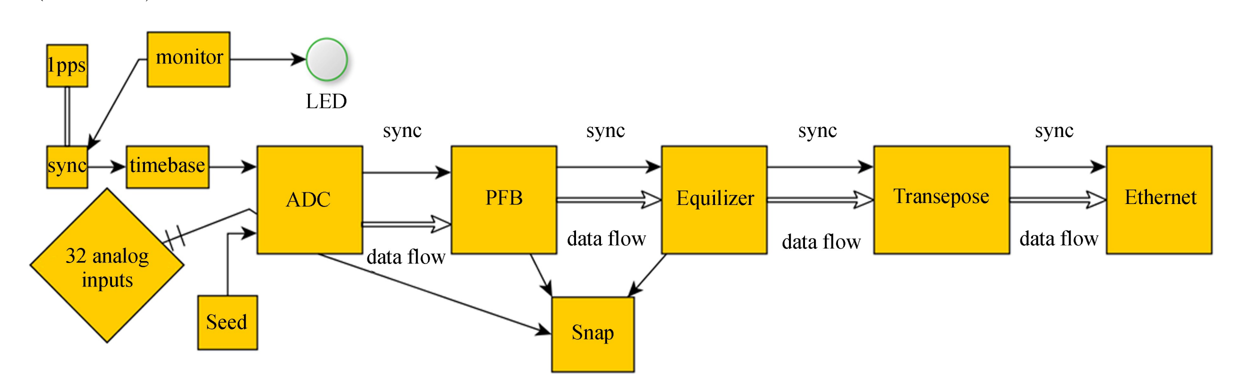


图 1　F-engine 整体功能架构示意图

Fig. 1　Functional framework of the F-engine

由于 X-engine 的运算负荷可根据实际情况调整，设计的架构可以通过调节转置和以太网模块的相应参数改变发送数据包的报文信息，从而适应后端 X-engine 运算节点数。

在运行中，ROACH2 由寄存器控制。ROACH2 将现场可编程门阵列中的寄存器映射到 PowerPC 中的 Linux 系统文件夹下，通过控制读写模型中寄存器的值控制模型的相应功能。ROACH2 还提供了远程连接的卡鲁望远镜阵列控制协议(Karoo Array Telescope Control Protocol，KATCP)功能，通过 ROACH2 服务器可以实现远程控制。该接口协议提供了相应的 Python、C、Ruby 库函数，通过 Ruby 编写了一套 F-engine 的控制程序。

2 F-engine 功能实现

2.1 模拟数字转换器和快速傅里叶变换功能的实现

F-engine 数据采集的功能由输入(Input)模块实现。输入模块内部主要功能是模拟数字转换器采集。在输入模块的子模块中，模拟数字转换器模块采用天文信号处理和电子学研究合作组提供的模块库中与硬件对应的 ADC16x250-8 模块，通过 ROACH 控制函数来控制采集卡。每块 ROACH2 配置 2 块 ADC16x250-8 采集卡，共完成 32 路采集工作。该模拟数字转换器采集完数据为 32 路 8 bits 数据流输入下一个模块。

为了抑制快速傅里叶变换之后的旁瓣效应，在进行快速傅里叶变换之前加入了多相滤波器模块进行多相滤波。即给每个信号值乘以一定的权重值。

$$S[v] = \sum_{n=0}^{L-1} w[n]x[n]e^{-2\pi ivn/L},$$

其中，S 是傅里叶变换后的光谱；L 是离散傅里叶变换长度；x 是输入时间序列；w 是权重函数即窗函数。加入的权重窗函数有汉宁窗、海明窗等多种形式，可以根据实际需求调节。ROACH2 软件系统中，快速傅里叶变换功能在多相滤波器模块中实现。使用由数字信号处理工具库提供的多相滤波器、快速傅里叶变换模块。

系统的快速傅里叶变换算法中，采用复数快速傅里叶变换运算，即每个快速傅里叶变换模块运算时将 4 路信号分别作为两个输入端的实部、虚部，最终做 2 路复数快速傅里叶变换运算产生 2 路复数信号。利用这种算法可以节省运算时间和现场可编程门阵列资源，每块 ROACH2 中只需 8 个复数快速傅里叶变换模块就可以满足前端模拟数字转换器输出 32 路信号的傅里叶变换。该模块输出时，每个快速傅里叶变换模块首先输出其中一个输入通道的频谱，然后输出另一输入通道的频谱，频谱中每个频点为 36 bits 复数，实部、虚部各为 18 bits。图 2 是频谱长度为 1 024 频点的一个快速傅里叶变换模块，处理 4 路输入信号，图中，pfb2、pfb3、pfb18、pfb19 为快速傅里叶变换模块的 4 个输入信号，fft2、fft3 为模块经过快速傅里叶变换运算之后的输出信号。

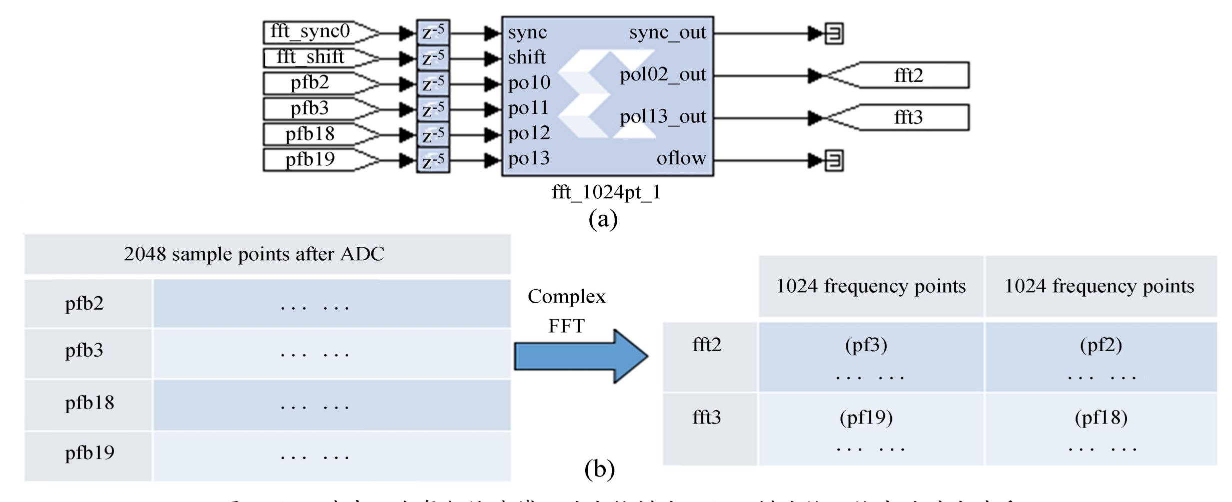


图 2 (a) 其中一个复数快速傅里叶变换模块；(b) 模块输入输出的对应关系

Fig. 2 (a) A complex-number FFT module; (b) The relation between data-input and data-output of the module

2.2　均衡器功能实现以及调节因子值范围的测定

在普通的接收机中，可能对某些频点响应强而对某些频点响应弱。在有限的动态范围下会影响结果精度。为了避免这个问题，在模型中加入了均衡器模块(Equalizer)进行频点通道补偿。在数据截断为 4 bits 之前，该模块使快速傅里叶变换输出的数据流通过一个范围调节过程，每个通道、每个频点加入一个自己的调节因子(EQ coefficient)简称 EQ 值。加入这个模块可以使接收机的频谱响应不随频率变化，即让接收机对每个通道、每个频点都有相同的响应。

均衡器模块中 EQ 值的大小对降低相关器带内不平整度很重要。通过输入高斯白噪声，根据输出通道自相关的均方根有效值确定调节因子值的大小。求均方根公式如下：

$$RMS = \sqrt{\frac{Autocorr}{\mathrm{Nint} \times 2}} ,$$

其中，*Autocorr* 是相关器 *N* 次积分后的自相关值：

$$\mathrm{Autocorr}[chan] = \sum_{Nint} \{[\mathrm{Rel}(chan, k) \times \mathrm{Rel}(chan, k)] + [\mathrm{Ima}(chan, k) \times \mathrm{Ima}(chan, k)]\} ,$$

其中，*N*int 表示 *N* 次积分，其值由 X-engine 设置所定；$k \subset (1, \mathit{N}\mathrm{int})$；*chan* 为通道编号。

经过均衡器输出的每个通道、每个频点是一个有符号位的实部、虚部各为 4 bits 的复数，所以得到的均方根值范围为(0~7)。根据天文信号处理和电子学研究合作组提供的经验值，当均方根有效值控制在 2 到 3 时可以得到比较好的线性度，在 5 到 7 时线性度比较差。调试的时候，调整每个通道、每个频点的调节因子值使得到的自相关均方根值在 2~3 的范围内，从而得到该通道、该频点合适的调节因子值。将天线对准无源天空，调节接收机的增益来调节输入相关器中信号幅度的大小。

2.3　转置(Transpose)功能实现

每个通道的数据流经过一次快速傅里叶变换运算后，只包含本通道的频谱，而 X-engine 做相关运算需要同一频点中，不同通道间进行交叉相乘，所以在 F-engine 发送数据前，需要将其重新排序拼接，这个过程称作转置(Transpose 或者 Corner Turn)。转置模块就是为了完成这个功能设计的。如图 3，以 32 通道输入、128 频点为例，最初数据是以通道-频点的方式排列，转置后数据以频点-通道的方式排列。

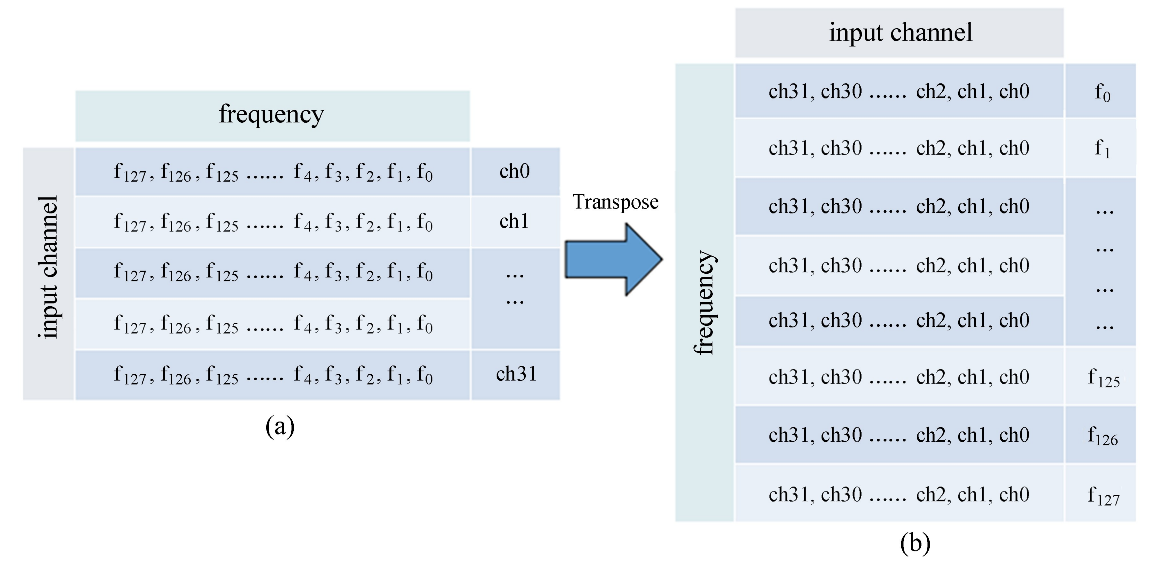


图 3　转置功能示意图。(a) 转置前数据流；(b) 转置后数据流

Fig. 3　Functional framework of the Transpose. (a) Original data flow; (b) Transposed data flow

每块 ROACH2 配备了一块万兆网卡，该网卡有 4 个万兆网口。为了合理地分配数据流，在转置模块内部设计了 transpose0、transpose1、transpose2、transpose3，4 个二级子模块，每块处理 1/4 频谱的转置，完成转置后分别送往 4 个万兆网口。图 4 是频率分辨率为 0.24 MHz(快速傅里叶变换长度为 512 频点)的转置模块的内部一级结构。图中每个二级子模块 transpose(0~3)的“dout”端口输出的数

据中，均包含 1/4 频段即 128 个频点的数据，转置后每个频点包含 ROACH2 板的 32 路通道数据。

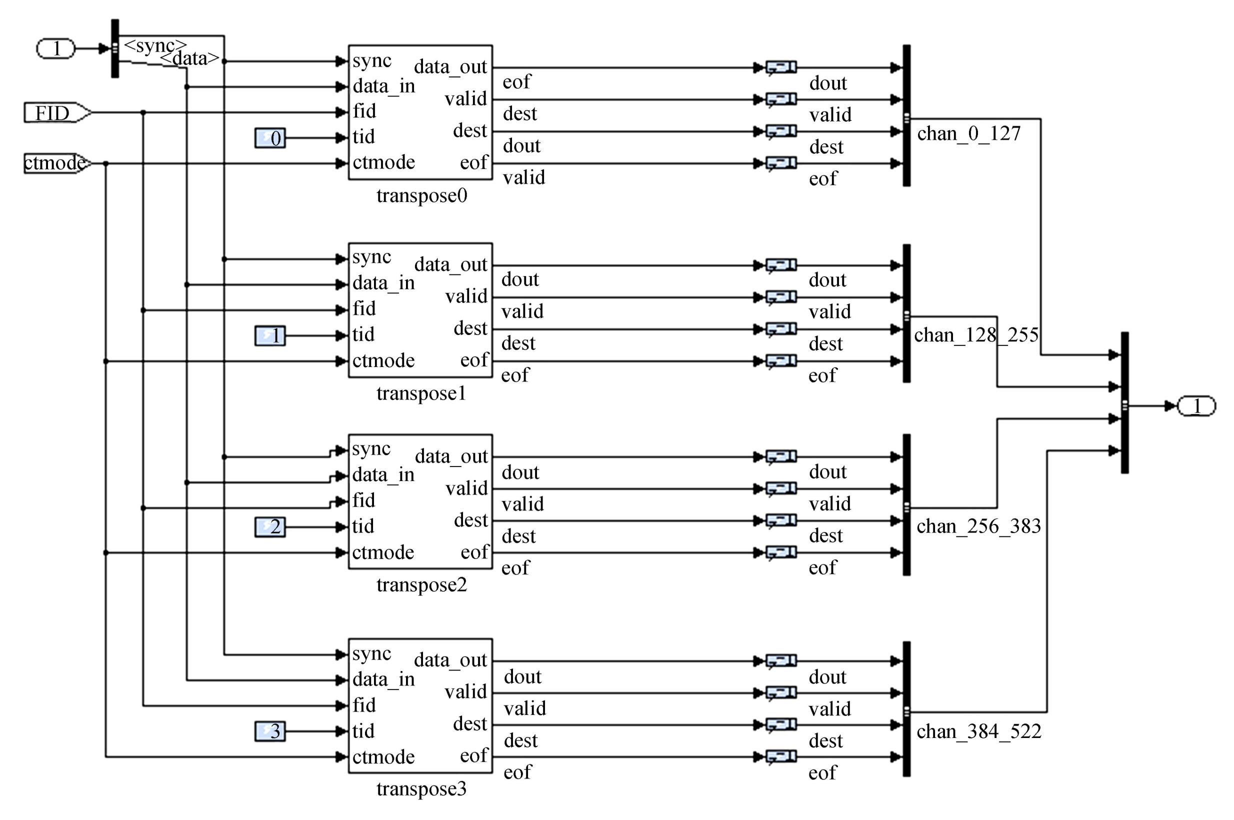


图 4 转置模块内部的一级子模块

Fig. 4 Structure of the first class sub-module in Transpose

转置二级子模块结构如图 5，其中每个二级子模块具有同样的架构。为了将通道数据按序写入数据包中，并完成频点-通道的转置工作，在每个转置二级子模块中，加入一个 Dual-port RAM 模块，通过控制读写地址的移位对数据流进行重新排序并完成转置功能。

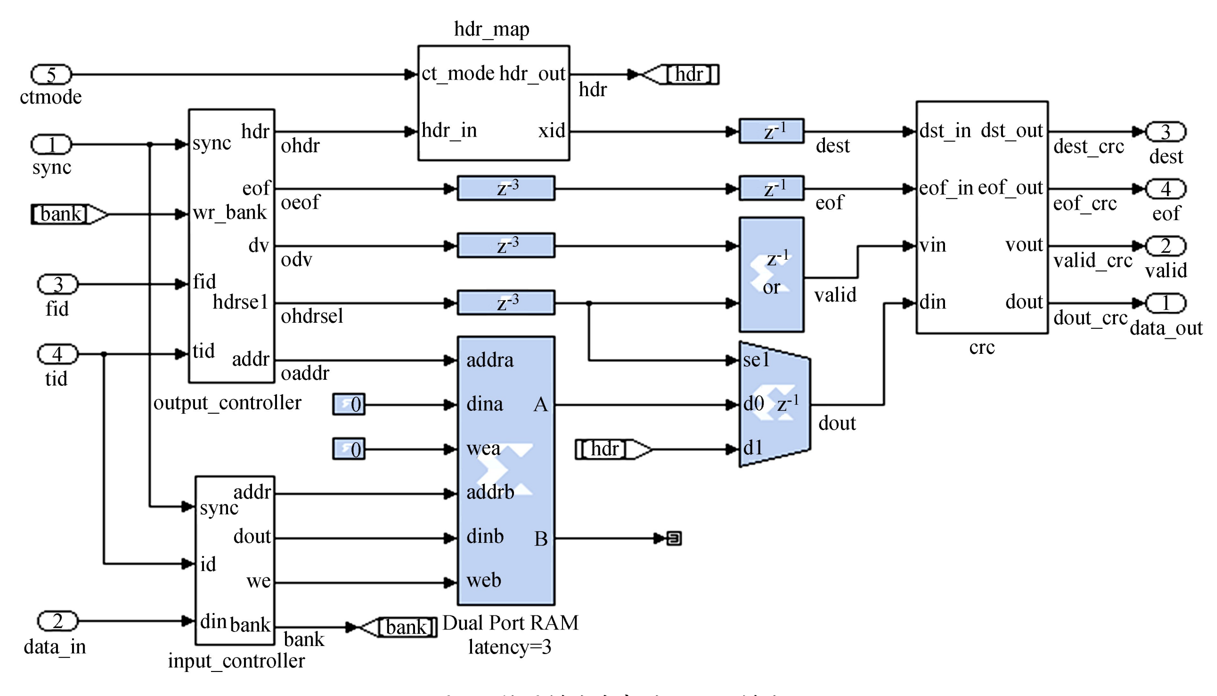
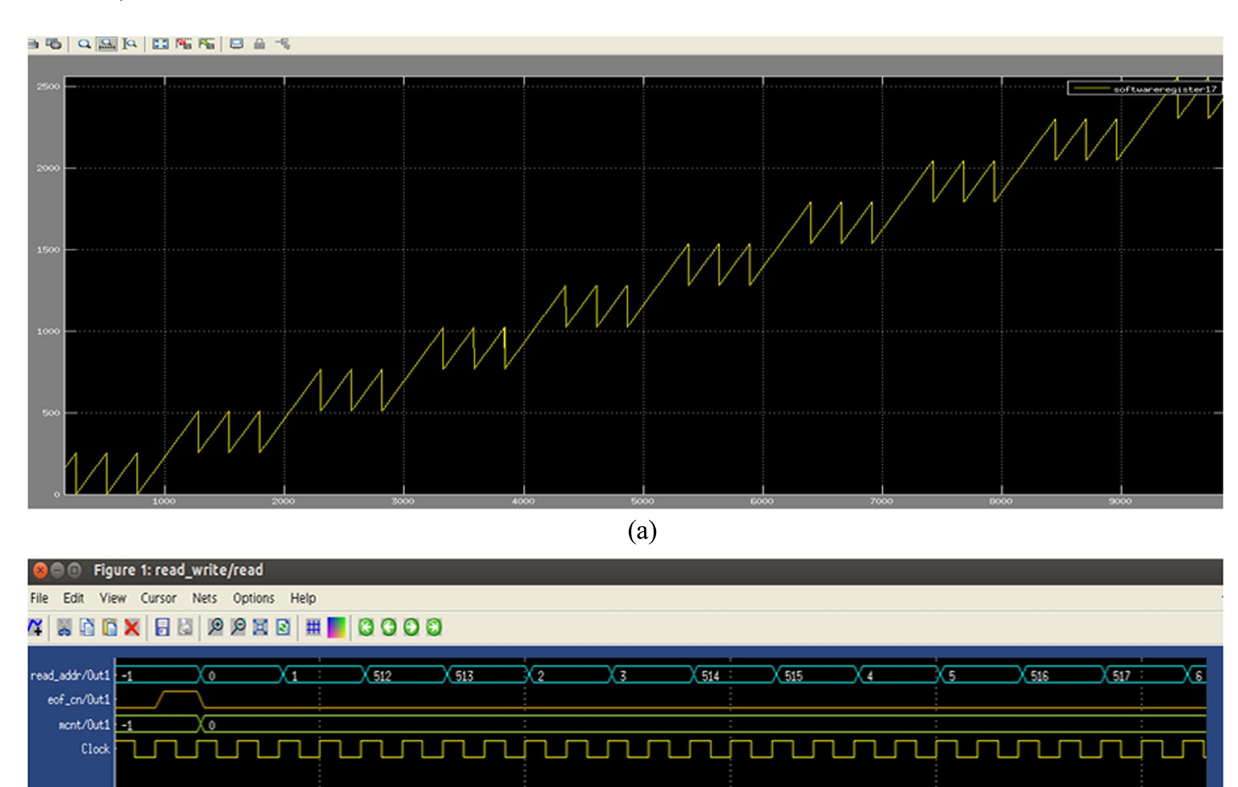


图 5 转置模块内部的二级子模块

Fig. 5 Structure of the second class sub-module in Transpose

图 6(a)是每个二级子模块内部 Dual-port RAM 模块写入地址随现场可编程门阵列时钟变化情况，可以看出，Dual-port RAM 模块的写入地址是从 0 开始重复 4 次，再继续增加。这样设计是因为有 4 个转置二级子模块，所以重复 4 次使得每个模块选择性地写入该频段的数据。在每个子模块中的 input_controller 模块内，由上层模块传入的“tid”信号数值来甄选是否将“data_in”端口传入的数据写入该子模块的随机存取存储器(Random-Access Memory，RAM)中。如果为该子模块所需频段，“we”信号为 1，“dout”端口传入的数据部分开始写入。



(a)

(b)

图 6　Dual Port RAM 读写地址输出。(a) 写地址输出随时间变化；(b) 读地址随时间变化

Fig. 6　The output of Writing and Reading address of Dual port RAM. (a) Writing address varied with time; (b) Reading address varied with time

由于前面多相滤波器模块进行复数快速傅里叶变换，实际上转置输入 16 个通道中的每个通道数据流先后包含了两个输入通道的数据流。图 6(b)为 Dual-port RAM 模块读地址输出，图中第 1 行为读地址，按照一前一后的交错地址读出随机存取存储器中写入的数据，这样可将后输出的通道移到同一位置，使得同一频点所有通道重新排序输出，如图 7。

每个数据包大小和其包头信息在转置模块中定义。根据不同的计数器和逻辑电路，在转置二级子模块的 hdr_map 模块中加入了目的 IP 信息

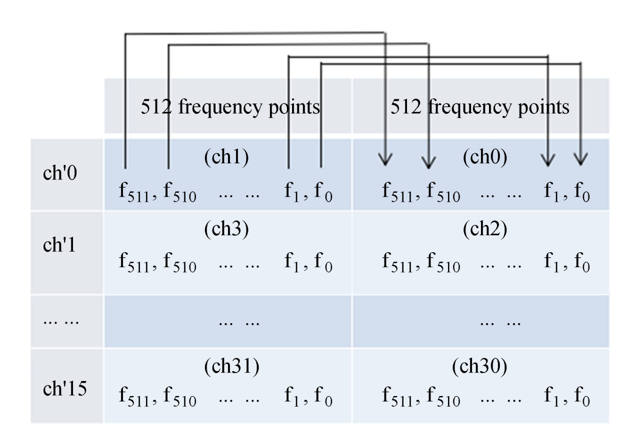


图 7　频点拼接示意图

Fig. 7　Schematic diagram of data assembly

和数据包包头信息。如图 5，目的 IP 信息由“xid”信号输出，最后在下一个网络模块中加入数据包头文件中。而其输出的数据包包头信息由“hdr”输出到一个判断选择器中，通过 output_controller 模块控制，在每个数据包正式内容前，先输入包头信息，接着再输入数据信息。

观测带宽为 125 MHz，根据观测需要，设计了快速傅里叶变换长度为 1 024 频点和 512 频点两种观测模式，即每个频点对应频率分辨率为 0. 12 MHz 和 0. 24 MHz。快速傅里叶变换模块由数字信号处理工具库提供，只需改变相应参数即可随意改变不同快速傅里叶变换长度。但由于不同快速傅里叶变换长度对应不同的频段发送，因此在转置功能实现中对应不同快速傅里叶变换长度，需要在转置模块中改变数据包包头信息，例如目的 IP 需根据快速傅里叶变换长度重新划分频段发送到相应 X-engine 单元。由于复数快速傅里叶变换后通道数非按序排列，Dual-port 随机存取存储器写入写出地址也要做相应改变。

2. 4 网络传输功能实现

在系统的内部网络中，不同的处理单元有各自的介质访问控制地址(Media Access Control，MAC)和互联网协议地址(IP)。该模块利用转置发出的 XID 为每个数据包加入不同的目的互联网协议地址和目的物理地址。每个网卡端口发送 ROACH2 板卡产生的 1/4 的频谱数据，根据后端 X-engine 运算需求，又将 1/4 频谱数据分为 8 段后加入目的互联网协议地址打包通过万兆交换机发送到 X-engine，即每个数据包将包含 1/32 频谱的数据。

为了节省万兆交换机中的网卡端口，探测再电离时期的精确阵列(Precision Array for Probing the Epoch of Re-ionizatio，PAPER)模型采用了一种循回的模式，将每台 ROACH2 中的两块网卡其中一块作为补偿交换机网卡端口。虽然每块 ROACH2 板只配备了一块网卡，但是有充足的交换机网卡端口，因此这里没有采用 PAPER 模型的网络模型，而是将数据包直接发给了交换机，由交换机通过物理地址寻址发往 X-engine，这种网络模型比 PAPER 采用的模型更简便，当路数变动时，只需要增加硬件设备将其连接到交换机并改变相应参数即可。

由于数据量大，传输速率高，而又不需要很强的安全性，本模型网络传输的数据包采用用户数据报协议(User Datagram Protocol，UDP)。数据包格式如表 1。

表 1 数据包格式

Table 1 The format of data packet

Header:	UDP Datagram	MCNT (6bytes)	FID (1byte)	XID (1byte)
f0	In0, In2, In3, In4, ... In28, In29, In30, In31			
f1	In0, In2, In3, In4, ... In28, In29, In30, In31			
...	...			
f31	In0, In2, In3, In4, ... In28, In29, In30, In31			

除了用户数据报协议报文，通过转置模块，在每个数据包里增加了包头部分用于记录每个数据包的信息。其中 MCNT 是一个计数器，用于 X-engine 识别收到数据包发送时刻；FID 用于区别不同的 ROACH2 板，从而在相关运算时换算出对应通道的编号；而 XID 则是记录数据包发送到 X-engine 的编号。

数据包内容按照频点-通道的模式排列，每个数据包含有 32 个频点，而每个频点包含了每块 ROACH2 所有 32 个采样通道。表中 In0-In31 表示 ROACH2 前端 32 路模拟数字转换器信号输入通道。数据包内将 8 个频点放到一个数据包中发送，增加了每个数据包的大小，这样可以充分利用万兆网络传输。每个通道进行完频点补偿后截断为 8 bits。为了验证网络传输，在数据包最后加了 8 bytes 的冗余校验位，这样除去 UDP 报文内容，每个数据包的大小为

$$N_f \times N_ch \times 8 + \mathrm{Header} + \mathrm{CRC} = 8\,208 \text{ bytes} .$$

而在以太网模块中，每个万兆网发一个数据包大约需要 8 192 ns，所以每个万兆网卡端口的传输速率约为 8 Gbps。

相关器 F-engine 还根据需求设计了两种频率分辨率模式，对应频率分辨率分别为 0. 12 MHz 和 0. 24 MHz。同时还设计了 RawData 模型用于检验模拟数字转换器的正确性。该相关器已经应用于天籁计划一期中 3 个柱面天线 96 个双极化馈源，共 192 路射电输出信号的干涉阵列，并对其性能进行了测试。

3　相关器的性能测试

3. 1　模拟数字转换器采样准确性测试

在系统设计中包括了一个原始数据(RawData)模式，可直接将模拟数字转换器采集的波形数据打包通过万兆网传到高性能计算机中存储，而不进行快速傅里叶变换和互相关等运算。这一模式可用于检验模拟数字转换器采样的正确性。输入一个 20 MHz 的正弦波给模拟数字转换器，其输出数据如图 8(a)。对上述数据进行拟合，算出其频率为 20. 000 1 MHz，与理论值基本一致。

3. 2　相关器输入输出线性度测试

为了评估相关器的输入输出线性范围，对相关器做了线性度测试。用无源的天空作为信号源，通过调整接收机变频器输入增益系数来控制输入信号的幅度，并输出不同幅度下的通道自相关值，画出线性范围。如图 8(b)，在输入信号-12 dbm~6 dbm 范围内有良好的线性度。

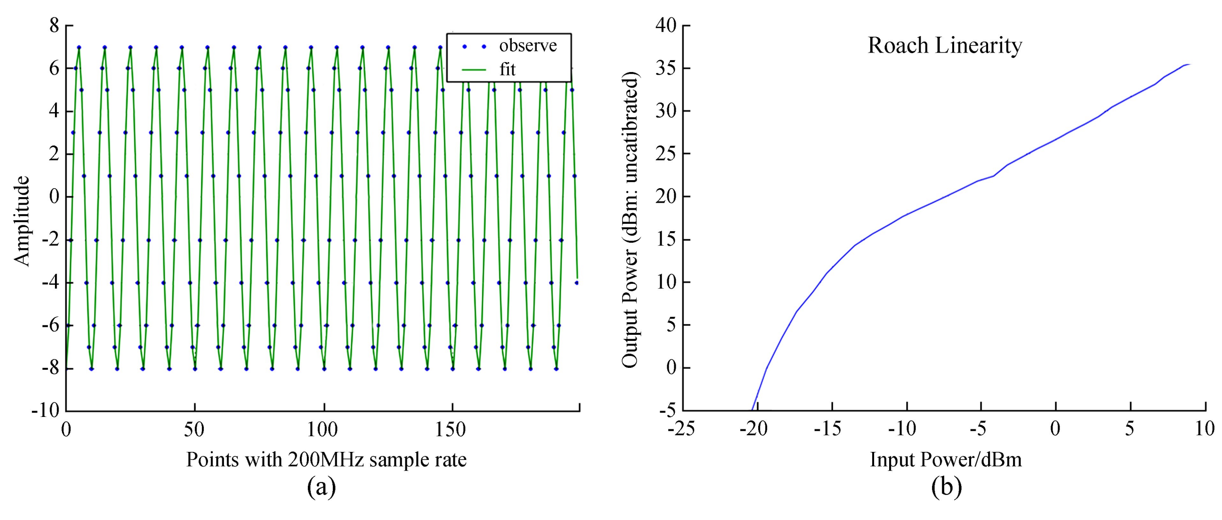


图 8　(a) 模拟数字转换器拟合；(b) 线性度测定

Fig. 8　(a) Fitting results of ADC ; (b) Linearity determination

3. 3　高斯白噪声模拟观测

在相关器的测试中，将同源信号输入不同的通道，并加入延时进行干涉实验，最终计算观测值与理论值的误差。在测试时，以模拟白噪声替代真实信号输入。输入两路信号可表示为 $Ae^{i(2\pi ft+\phi)}$ 与 $Ae^{i(2\pi f(t+\tau)+\phi)}$，其中，$\tau$ 是两面天线所接收到信号的延迟。

相关相乘后的能见度为

$$V = Ae^{-i(2\pi ft+\phi)} * Ae^{i(2\pi f(t+\tau)+\phi)} = Ie^{i2\pi f\tau}.$$

其相位为

$$\phi = 2\pi\tau f.$$

如果时延 τ 是固定的，则相位随频率 f 单调递增，斜率为 $k=2\pi\tau$。实际上相位随频率在 $-\pi$ 到 π 之间周期变化。

用不同的噪声源进行了两个实验：实验一：噪声源采用 ROACH2 产生的模拟数字噪声替代模拟数字转换器采样直接输入后端。利用现场可编程门阵列中的时延模块产生延时模拟干涉阵输入信号。

其相位变化如图 9(a)。加入 2.5×10^{-8}s 的时延，理论上相位随频率变化斜率为

$$k = 2\pi\tau = 2 \times \pi \times 2.5 \times 10^{-8} = 1.571 \times 10^{-7}\text{rad/Hz} .$$

观测得出的变化率拟合值为 1.561×10^{-7}rad/Hz，误差为 0.6%。变化率拟合如图 9(b)。

实验二：用噪声发生器产生真实噪声输入模拟数字转换器。利用噪声源到模拟数字转换器之间射频线长度差产生时延模拟干涉阵输入信号。采用射频线长差为 7.5 m，测得的信号在射频线中传播速度约为 0.7 倍光速，该值在合理范围内。

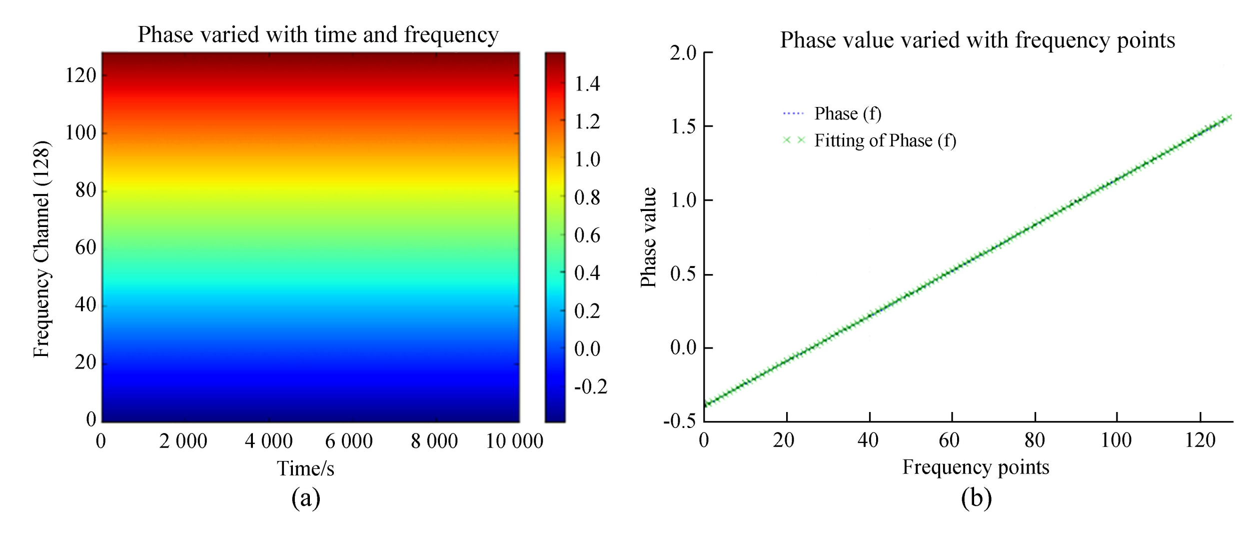


图 9 (a) 加入时延和模拟噪声后相关器输出两路信号的干涉相位图；(b) 某时刻相位随频率变化的观测和拟合图

Fig. 9 (a) Interfere phase of two signals varied with time and frequency after adding simulation noise and time delay; (b) An output of phase slope and its fitting result

3.4 射电源观测测试

使用相关器结合 2 面天线和接收机等设备对太阳，Cygnus A 射电源进行了干涉观测，通过离线成图可以明显看到由射电源运动引起的干涉振幅幅度的变化和相位条纹的变化。还利用另一套由中国科学院自动化所开发的相对成熟的相关器同时对 Cygnus A 进行观测，将得到的数据进行比较，结果基本吻合，详情参见文[7]。

4 总 结

本套相关器 F-engine 模型实现了数据采集、快速傅里叶变换、转置、网络分发等功能。在此基础上，还加入了 Raw_Data 模型及其控制程序，可将采集的数据直接存盘，方便检查模拟数字转换器的正确性。相关器易于根据通道数扩展，对于干涉阵列增加，无需开发新的模型，只需要改变相应参数即可。

对于运算精度要求不断提高，F-engine 的运算负荷将增大。但随着 CUDA 编程的日益完善，将来可以通过图形处理器并行运算完成快速傅里叶变换，未来的相关器可将 F-engine 中的快速傅里叶变换移交到图形处理器后端进行，而 ROACH2 只作为采集卡和网络分发的工具。ROACH2-GPU 这种架构的相关器已经慢慢被越来越多的天文技术人员接受，它的易扩展、易开发的优势为射电天文的发展增加了新的活力。

参考文献：

[1] Thompson A R, Moran J M, Swenson Jr G W. Interferometry and synthesis in radio astronomy [M]. New York: John Wiley & Sons, 1986.

[2] Harris C. A parallel model for the heterogeneous computation of radio astronomy signal correlation [D]. Australia: the University of Western Australia, 2009.

[3] 陈学雷. 暗能量的射电探测天籁计划简介 [J]. 中国科学：物理学 力学 天文学, 2011, 41(12): 1358-1366.
Chen Xuelei. Radio detection of dark energy-Tianlai project [J]. Scientia Sinica: Physica, the Mechanica & Astronomica, 2011, 41(12): 1358-1366.

[4] Parsons A R, Backer D C, Bradley R F, et al. The precision array for probing the epoch of reionization: 8 station results [J]. Astronomical Journal, 2009, 139(4): 1468-1480.

[5] 裴鑫, 李健, 陈卯蒸, 等. 基于 ROACH 的微波全息法相关机设计 [J]. 天文研究与技术, 2015, 12(1): 54-62.
Pei Xin, Li Jian, Chen Maozheng, et al. A design of a digital correlator for microwave holography based on the ROACH platform [J]. Astronomical Research & Technology, 2015, 12(1): 54-62.

[6] 徐永华, 李纪云, 张颖倩, 等. 相干消色散脉冲星观测系统的研究 [J]. 天文研究与技术, 2015, 12(4): 480-486.
Xu Yonghua, Li Jiyun, Zhang Yingqian, et al. A study of a pulsar observation system using the coherent de-dispersion technique [J]. Astronomical Research & Technology, 2015, 12(4): 480-486.

[7] 汪群雄, 牛晨辉, 田海俊, 等. 基于 ROACH2-GPU 的集群相关器研究——X-engine 模型的设计与实现 [J]. 天文研究与技术, 2016, 13(2): 219-227.
Wang Qunxiong, Niu Chenhui, Tian Haijun, et al. A research on the ROACH2-GPU-cluster-based correlator-the design and implementation of an X-engine model [J]. Astronomical Research & Technology, 2016, 13(2): 219-227.

Research on the ROACH2-GPU-Cluster-Based Correlator
——Design and Implementation of an F-engine Model

Niu Chenhui[1,2], Wang Qunxiong[1,3], Zheng Xiaoping[2], Tian Haijun[1,3],
Wu Fengquan[1], Li Jixia[1], Chen Xuelei[1], Hao Jie[4]

(1. National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China, Email: hjtian@lamost.org;
2. Central China Normal University, Wuhan 430079, China; 3. China Three Gorges University, Yichang 443002, China;
4. Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Correlator plays an important role in radio interferometry astronomy. Traditional correlator is achieved with FPGA or ASIC. It calls for long development cycle and its flexibility is unsatisfactory. At present, more and more radio astronomy arrays adopt generic architecture correlator with FPGA system and GPU. In order to meet the demands of the ‘Tianlai’ project, we developed a new correlator based on ROACH2 and GPU cluster. We divided the workflow of the correlator into two parts: we let the high performance GPU undertake the CMAC (conjugate multiply and accumulate) and only left the ADC and FFT for the F-engine. This correlator has high flexibility and it is easy to control. It has been applied to the Tianlai project.

Key words: Correlator; F-engine; ROACH2; Tianlai project